Vector Animation: Web-Based Software Training On Demand

Carol J. Scott
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive
Pasadena, California, USA 91109-8099
(818) 393-0551
Carol.J.Scott@jpl.nasa.gov

Abstract--At the Jet Propulsion Laboratory, training developers are creating vector animations to instruct engineers in the operation of mission software. These movies simulate most aspects of software operations providing a safe learning environment for flight teams. The technology is readily available as a COTS (Commercial Off The Shelf) program called Flash. Flash movies are low cost, highly productive, easily modified and maintained, and deliverable via our Intranet. Animations are excellent vehicles for delivering training across the Internet and they present it in a format that it is easy to understand and enjoyable to use. Since it is available any time of the day or night, users adopt their own pace, digesting information at a rate suitable for their learning style. There is no delay between actions, the user chooses a path but can change direction or return instantly. Action-oriented Flash movies are very powerful, relatively small, quick to load, and offer tremendous potential for increased productivity. One obvious benefit would be a reduction in the need for costly and repetitive travel.

TABLE OF CONTENTS

1. INTRODUCTION

The Customer Integrated Services Group of the Mission Execution and Automation Section, Information Technologies and Software Systems Division at JPL (Jet Propulsion Laboratory) is currently developing vector animations for delivering mission software training to our customers. "Customer" refers to mission flight teams, remote science investigators, and Deep Space Network personnel who use JPL's mission and ground system software.

Training is most effective when it is willingly consumed by users for immediate application to their tasks. Animations (in this case Flash movies) provide engineers the opportunity to "use" our subsystems by presenting simulations of graphical user interface (GUI) actions and reactions, with respect to user inputs. Movie simulations provide a safe environment for users to learn and practice their skills by, not only simulating GUI operation, but by "invoking" nested applications as the actual software would. There is no delay between actions, the user chooses a path and can change direction or return instantly.

Flash movies can easily branch to other aspects of system operation. The developer can provide a main menu of options that includes a range from high level overviews to very specific low level demonstrations and interactions, then on to optional methods for achieving objectives.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

2. WEB-BASED DELIVERY

The Internet is a great tool, but it can be a time-consuming quest if answers are buried in a maze of nested text objects where users cannot guess the proper keywords. When users need answers "now", an on-line game of Clue will not ease their frustration.

Flash movies are excellent vehicles for delivering training across the Internet and they present it in a format that it is easy to understand and enjoyable to use. A movie operates and reacts as the actual software does, you navigate to the location of interest, and let the movie guide you through the process. You visually "see" what happens as a result of your inputs or actions, and text descriptions annotate the action as you progress through the steps.
Action-oriented Flash movies are powerful, relatively small, and quick to load. Because the Flash player is

already "plugged" into most of the newer Internet browsers, the code that runs (interprets) the movie is already resident on each machine; only the movie content is downloaded to the user. During the development of a Flash movie, each image (or partial image) is imported once, potentially used many times within the script, then downloaded to the user when invoked from the user's Internet browser.

When customers select a Flash movie from the training web site, they operate at their own pace, digesting information at an ideal rate for their learning style. Processes are precisely displayed in pictures or sequential flows using various animation and navigational techniques. An on-line training module can be invoked from a browser window for immediate usage (on-demand) while the real software runs in another window.

3. Animation Technology

Pictures Describe the Players

Pictures can take almost any form from photographs to pencil sketches. You can digitize most images to convey the likeness of your subject, although the large file sizes of photo-quality images translate into slow response time on downloads. Vector graphics offer an alternative to photographs; photographs can be converted to vector graphic format.

Simplistically, vector graphics are lines with fill. They can be easily taken apart and reconstructed in any configuration. Most pictures can be easily understood without the brilliance and resolution of a photo quality image. Millions of colors can be removed from pictures, while the clarity of the meaning is maintained in a significantly smaller format.

3.2 Motion Tells the Tale

By adding motion, the pictures can interact with each other in a meaningful way. Flash allows you to create the illusion of movement and shape-shifting by employing the technique of "tweening." The developer provides the first image object, the last image object, and a few critical mid-stream changes, then tweening fills in the rest to create a smooth transition between strategically placed images. The effect resembles simple morphing (this is not a true morphing technique.) Tweening simply moves an object from point A to point B, while controlling its size and orientation. (True morphing would transition selected pixels in a picture until they took on the appearance of pixels in a second picture.)

Each object is manipulated independently of other objects through the use of layers. A layer assigns visual priority to an object. An object at the top of the layer stack will appear to cover an object at the low end of the layer stack. For instance, a story can be told over a stationary background object. The background could be a single, low lever layer.

As the cast of objects is built, each significant object of motion is maintained on a separate layer. In the instance where objects always move together, they can be managed on the same layer. Same-layer objects can change places dynamically, front to back, where layered objects cannot, unless they are maintained on multiple layers.

Motion Over Time Reveals the Purpose

Flash movies operate on a time line of frames, where object layers move together relative to their position in any given frame. For instance, an object may move along one path over a sequence of frames while another object, moving through the same sequence of frames, takes a different path and performs a different set of actions. The greater the number of incremental steps (frames) included, the smoother the motion appears.

4. Benefits

In developing software training, the closer to the real user environment, the more beneficial the training experience. Software training may be impractical in the real environment and software may not be portable enough to use anywhere. Platforms and versions often become an issue in software training. Even a dedicated training environment suffers occasional "blackouts" after software upgrades, reconfigurations, version releases, and so on. Software simulations can eliminate these types of problems.

Animations are available over the web, any time of the day or night. Flash movies convey processes, procedures, and detail in an enjoyable way; they can be designed to be self-testing and diagnostically astute at determining individual learning needs. Information can be collected for analysis when desired.

5. Why Animation

As spacecraft become smaller, more intelligent, and more numerous, hardware and software support structures become increasingly complex. One challenge is to develop enhanced productive methods of training to satisfy the increasing development load, while being constrained by existing training resources. Most engineers, if given the choice, would vote to spend their budgets on research and development. Training, for them, is often something that happens...at the end, after delivery. Many engineers are so busy that they would almost rather attempt to figure it out for themselves rather than take time for a training class. Many engineers are self driven in an effort to solve a complex problem. They may work all night to solve a problem or develop an enhancement. When they encounter a problem in the middle of the night is when they want their training.

Another major challenge is keeping our bright and determined flight team engineers, science representatives, and operations personnel together long enough to train them. Getting them together is difficult, given their scheduled deliveries, task meetings, and other project commitments. Choreographing a training session for a group of engineers is like...herding mosquitoes. They may start off at the same place, but each has his or her own agenda for learning.

Workstation training has fluctuated from large group hands-on sessions to small group, or even one-to-one hands-on sessions. What started as a full set of courses presented over five consecutive days, has evolved into a just-in-time scenario where the user determines the schedule, material sequencing, and course content based on current need.

Engineers require more than a list of instructions to follow. They need to understand the entire process. If problems arise, they need to understand why and how to work around the difficulties, perform rudimentary diagnostics to determine the source of the problem, and understand the outcome of each alternative decision. With multiple subsystems and levels of communication, various dataflow conditions and data source connections, multiple storage devices, and clocks ticking in multiple formats, there is no single software recipe for success.

Automating software training for our engineers has been a goal for years. We have made several attempts at developing computer-based training; however, mission software is extremely intuitive and requires significant experience and background knowledge. It is difficult to code the subtleties into an on-line package.

6. FAILED ATTEMPTS

The first attempt back in the early 1990s involved a program that was used in developing procedural training for the Army. We attempted to adopt it for software training because it was very thorough; however, because it was so thorough, it was tedious and difficult to implement, given the speed at which software deliveries were produced. The next attempt used a Perl shell on a UNIX workstation. The Perl driver called up a sequence of window dumps narrated by voice-overs. It was relatively easy to build and very inexpensive, but was too labor intensive to be a serious contender.

We looked next at Authorware, a program with an excellent reputation for developing, managing, and delivering training, and which supported our Macintosh development platform. We also purchased Director, which enabled us to develop software tool scenarios, which were to be included in the Authorware production. Unfortunately, we had only minimal resources to perform both ongoing training and new development. Although supporting ongoing training efforts won, the initial results were quite dramatic. Our vision was a program that would perform end-to-end training for all project personnel , focusing on the most commonly used software tools. We couldn't develop and keep up with our training schedule, so we ended up using the test modules as on-line examples of what might have been. Our design had been too comprehensive.

Flash started as a toy, purchased privately for personal use. It showed great promise immediately, so it was purchased as an experiment for training development. This time we limited our designs to a single subsystem per movie.

Vector animations are best described by looking at the design and development processes, analyzing the components, and assessing the benefits to the user population.

7. HOW FLASH MOVIES WORK

A Flash movie is analogous to a theater production. In a play, the story is usually acted out within the confines of a stage. If the location of the action in the story changes, a scene change takes place where the curtain is drawn and new scenery installed. Scene changes may occur several times during the play. Sometimes, however, your focus is drawn towards an aside or action vignette temporarily localized on a small portion of the stage. When this happens, even though the background scenery does not change, your attention is carefully focused on the action and controlled via lighting or other special effects. The play runs its course in a predetermined amount of time, where each actor has a prescribed amount of time to convince you of his or her purpose.

Flash movies are also developed within the confines of a "stage" where stage size can be preset by the designer, or dynamically resized by the browser. A movie is composed of one or more scenes, where each scene is built upon a timeline of individual frames. A frame is a picture of your stage at any given point in time. Action is laid upon a frame in layers. The objects in a layer are all static to one another. Action is produced as the elements of one layer change in relationship to the objects in all the other layers at the same frame

Scenes are sequential in nature, but one scene can call another scene to replace it at any point during the action of the scene. Scenes can also call movie clips that play within the action of the current scene. Movie clips are like scenes because they are built upon timelines, however, movie clips are not sequential in nature. Movie clips can appear or disappear at any time without replacing the timeline of the current scene.

8. Development

JPL uses graphical user interfaces for communicating with most mission software subsystems. With complex processing concepts, the data flow structure that relates subsystems to each other can be used as a guideline template for sequencing the GUIs (showing data flow as a function of operator actions.) Some subsystems perform many functions involving multiple programs, where each program may supply its own GUI (or set of GUIs.) Data flow structure is a natural basis for subsystem training development and provides a clear organizational path for GUI hierarchies.

When designing a GUI simulation, the focus is on data flow, where it came from, where it is going, and how or why the user will interact with it. It is important to understand why the user might choose one GUI over another when multiple methods are available. This will provide insight for offering usage diagnostics on the back end of development.

Design the stage in terms of where the action takes place. Size the stage to accommodate the largest GUI, or group of multiple GUIs. When a single graphical user interface is used, consider creating the presentation within a single scene. When additional GUIs are called from the main GUI, consider their size and function when deciding how to introduce them into the action. A large GUI that would overwhelm the original GUI can be introduced as a new scene, replacing the main GUI on the stage. When a GUI is small and called to affect the action of the main GUI on the stage, design a movie clip to play over the main GUI at the appropriate moment, then disappear.

Flash software simulations should convey the realistic experience of each tool. Starting with the actual software as the base, save TIFF (Tagged-Image File Format) files of the GUI in its different stages of operation. Some GUIs look the same throughout the process except for text, buttons, and menus. Other GUIs may shrink or expand with additional sections based upon user input or stage of operation. The movie can add or subtract from the GUI content by overlaying changes in GUI appearance.

Next we open our TIFF file within Fireworks, another Macromedia product that reduces file size, and export it as a GIF (Graphics Interchange Format) file. Upon export, we specify web-safe colors and reduce the number of colors to a minimum.

Within Flash, it sometimes helps to turn off anti-aliasing to help clarify small text within the image...or cover it over with fill and overwrite new text.

User Interactions

We use buttons or pull-down menus that imitate the actions of their counterparts on the real GUIs. In fact, this is the way in which we transition to other scenes or introduce movie clips to play in front of the main GUI.

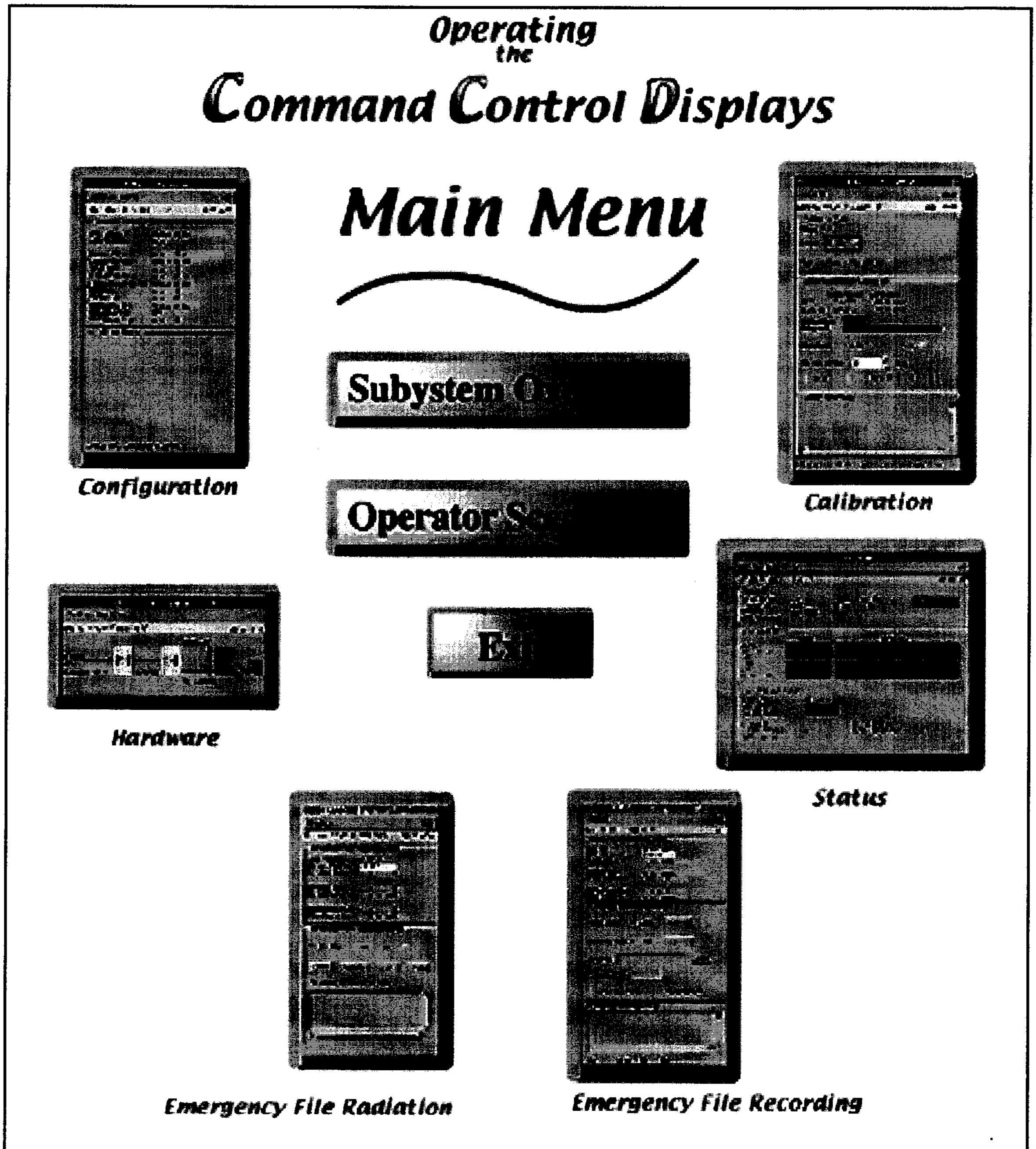
Buttons

Buttons are very interesting creations. You can use standard Flash buttons, make your own eye-catching creations, or use the real buttons appearing on the GUIs. I prefer the real ones. The more closely a button appears to resemble the original, the more realistic the experience. One method for adding depth to a button image on the GUI is the use of invisible buttons with shaded sides. This lets the user respond to the appearance of the real button, at the same time giving it depth with an unmistakable button quality.

Buttons are objects that can be used any number of times in a movie, and each instance of a button is associated with a specific set of actions. Buttons have three separate states (up, over, down) where each state may have its own appearance. The button's change in appearance helps identify it as a button.

Menus

For pull-down menus, we used a series of frames to isolate the action and reaction of the menu options. Take care to notice any buttons that you cover over with a pull-down menu. The buttons remain active and



Menu and GUI Selection Buttons

may interfere with the action of the menu options. We solved this problem by removing the button frames while the menus were visible.

Data

Data as it displays in real time is difficult to simulate. To provide absolute control over data flow, we chose to physically add data layers and change the values in selected frames of our movie clips and scenes.

Variables

We use variables to assign processing states to subsystems that communicate those states to other GUIs. There may be multiple GUIs that have the ability to change the state of a subsystem and additional GUIs that report on it.

Feedback

Variables can store data about the training session or as requested of users. Forms then collect these variable values and submit them back to Training log files for diagnosing user skills or tracking metrics.

Wait State

A favorite wait state is a movie clip with a nearly invisible clock that spins its hands once in approximately four seconds. At the end of each rotation, it checks a variable containing the number of times the hands should spin. The value is set when the wait state is invoked; the clock disappears when through.

9. Maintenance

Flash is a powerful and flexible program. Maintenance will depend upon your ability to clearly describe and label all elements of the animation. Use labeling capabilities provided and set up a traceability map. Modifying the animation is simple; its making sure that one modification does not affect an unseen element. Movie clips are especially vulnerable. Don't move objects in a movie clip without a complete understanding of how it plays while partially covering a scene.

Comment and use descriptive labels. Map out scenes and movie clip structures. Clearly identify all variables, actions, and layers.

10. User Acceptance

To assess user interest in on-line animated training, a small demonstration module was installed on the Training web page and announced to a small user population. The number of web site hits increased dramatically, some returning frequently to check on updates. One mission has requested this type of training for their remote science community.

Flash technology is relatively low cost, highly productive, easily modified and maintained, and best of all...deliverable via our Intranet. Newer browsers already contain the required player for many operating systems (at this time a Solaris solution is available, but unsupported), and the Flash program itself generates an html driver. Flash movies offer tremendous potential for increased productivity. We have only begun to identify its possibilities.

11. Conclusions

Movie development is especially rewarding as you see immediate results of your efforts. To begin a development, start with a simple project to get a feel for the capabilities of Flash. Develop a style (look and feel) that defines your purpose and stick with it as an overall theme throughout your productions. It will save time throughout the development process. Comment, comment, comment! Enlist volunteers to test your ideas and provide you with feedback...then listen to them!

Biography

Carol Scott is a Lead Information Technology Specialist at the Jet Propulsion Laboratory where she has been developing software training for mission flight teams for the past nine years. She authored three papers on training development. She has BS in Computer Science from Chapman College and a BA in English from the University of Nebraska at Omaha.